19980707 128

IC IP-98
Chicago, IL

IMAGE HALFTONING and INVERSE HALFTONING for OPTIMIZED DOT DIFFUSION

Murat Meşe and P.P. Vaidyanathan
Department of Electrical Engineering 136-93,
California Institute of Technology, Pasadena, CA 91125 USA,
Phone (626) 395-4681 Fax: (626) 795-8649
E-mail: mese@systems.caltech.edu, ppvnath@sys.caltech.edu

ABSTRACT

The dot diffusion method for digital halftoning has the advantage of parallelism unlike the error diffusion method. However, image quality offered by error diffusion is still regarded as superior to other known methods. In a recent paper we showed how the dot diffusion method can be improved by optimization of the so-called class matrix. In this paper we first review the dot diffusion algorithm and the optimization of the class matrix. A method for inverse halftoning of dot diffused images is then proposed. The method uses wavelet decomposition to eliminate the halftoning noise and does not make use of the knowledge of the class matrix.

1 INTRODUCTION

Digital halftoning is the rendition of continuous-tone pictures on displays that are capable of producing only two levels. There are many good methods for digital halftoning: ordered dither [3], error diffusion [4], neural-net based methods [2], and more recently direct binary search (DBS) [10]. Ordered dithering is a thresholding of the continuous-tone image with a spatially periodic screen [3] . In error diffusion [4], the error is 'diffused' to the unprocessed neighbor points.

Ordered dithering is a parallel method, requiring only pointwise comparisons. But the resulting halftones suffer from periodic patterns. On the other hand error diffused halftones do not suffer from periodicity and offer blue noise characteristic [11] which is found to be desirable. The main drawback is that error diffusion is inherently serial, e.g., to get the halftoned value of the last pixel, all of the remaining points should be processed. Also there occur worm-like patterns in near mid-gray regions and resulting halftones have ghosting problem [5]. Mitsa and Parker have optimized ordered dither matrix [9] for large size like 256x256 to get the blue noise effect. This is a compromise between parallelism and image quality.

The dot diffusion method for halftoning, introduced by Knuth [5], is an attractive method which attempts to retain the good features of error diffusion while offering substantial parallelism. However, surprisingly, not much work has been done on optimization of the so-called class matrix. In [8] we showed that the class matrix can be optimized by taking into account the properties of human visual system (HVS). The resulting halftones are of the same quality as for error diffusion. Since dot diffusion also offers increased parallelism, it now appears to be an attractive alternative to error diffusion.

In this paper, we first review the improved dot diffusion algorithm of [8], and then address the inverse halftoning problem. Inverse halftoning has a wide range of applications such as compression, printed image processing, scaling, enhancement, etc. In these applications, operations can not be done on the halftone image directly, and inverse halftoning is mandatory. A simple yet efficient algorithm for inverse halftoning of dot diffused images is proposed and compared to other methods.

2 REVIEW OF DOT DIFFUSION

The dot diffusion method for halftoning has only one design parameter, called the **class matrix C**. It determines the order in which the pixels are halftoned. Thus, the pixel positions (n_1, n_2) of an image are divided into IJ classes according to (n_1 mod I, n_2 mod J) where I and J are constant integers. For example, Knuth used a class matrix of size $I = J = 8$, and there were 64 class numbers in that class matrix [5]. Let $x(n_1, n_2)$ be the contone image with pixel values in the normalized range $[0, 1]$. Starting from class $k = 1$, we process the pixels for increasing values of k. For a fixed k, we take all pixel locations (n_1, n_2) belonging to class k and define the halftone pixels to be

$$x_h(n_1,n_2) = \begin{cases} 1 & \text{if } x(n_1,n_2) \geq 0.5 \\ 0 & \text{if } x(n_1,n_2) < 0.5 \end{cases}$$

We also define the error $e(n_1, n_2) = x(n_1, n_2) - x_h(n_1, n_2)$. We then look at the eight neighbors of (n_1, n_2) and replace each contone pixel with an adjusted version for those neighbors which have a higher class number (i.e., those neighbors that have not been halftoned yet). To be specific, neighbors of x(i,j) with higher class numbers are replaced with

$$x(i,j) + 2e(n_1,n_2)/w \quad \text{(for orthogonal neighbors)} \quad (1(a))$$

$$x(i,j) + e(n_1,n_2)/w \quad \text{(for diagonal neighbors)} \quad (1(b))$$

where w is such that the sum of errors added to all the neighbors is exactly $e(n_1, n_2)$. The extra factor of two for orthogonal neighbors (i.e., vertically and horizontally adjacent neighbors) is because vertically or horizontally oriented error patterns are more perceptible than diagonal patterns.

The contone pixels $x(n_1, n_2)$ which have the next class number $k + 1$ are then similarly processed. The pixel values $x(n_1, n_2)$ are of course not the original contone values but

[1]This work was supported by Office of Naval Research Grant N00014-93-1-0231.

the adjusted values according to earlier diffusion steps (1). When the algorithm terminates, the signal $x_h(n_1, n_2)$ is the desired halftone. In IJ steps the algorithm will complete the halftoning process.

Usually an image is enhanced [5] before dot diffusion is applied. For this the continuous image pixels $C(i,j)$ are replaced by $C'(i,j) = \frac{C(i,j)-\alpha\bar{C}(i,j)}{1-\alpha}$ where $\bar{C}(i,j) = \sum_{i-1}^{i+1}\sum_{j-1}^{j+1} C(i,j)/9$. If $\alpha = 0.9$ then

$$C'(i,j) = 8C(i,j) + C(i,j) - \sum_{0<(u-i)^2+(v-j)^2<3} C(u,v).$$

This algorithm is completely parallel requiring 9 additions per pixel, and no multiplications.

3 OPTIMIZATION OF CLASS MATRIX

Knuth introduced the notion of **barons and near barons** in the selection of his class matrix. A baron has only low-class neighbors, and a near-baron has one high class neighbor. The quantization error at a baron cannot be distributed to neighbors, and the error at a near baron can be distributed to only one neighbor. Knuth's idea was that the number of barons and near barons should therefore be minimized. He exhibited a class matrix with two barons and two near barons. The quality of the resulting halftones are still inferior to error diffusion because of periodic patterns similar to ordered dither methods (see Fig. 6). In our experience, the baron/near-baron criterion does not appear to be the right choice for optimization as explained in [8]. In Sec. 3.1 we introduce a different optimization criterion based on the HVS, and show that the image quality is significantly improved, though the class matrix does not minimize barons.

3.1 Objective Function Based on Blue Noise

It has been observed in the past that the error in a good halftone should have the **blue noise** property [11]. This means that the noise energy should mostly be in the high frequency region where it is known to be less perceptible. We showed in [8] how to incorporate blue noise characteristics into the class matrix optimization:

Let $x_h(n_1, n_2)$ denote the halftoned version of a constant gray image $x(n_1, n_2) = g$ where $0 \leq g \leq 1$.Typically, the dark pixels are spatially distributed with a certain average frequency f_g called the **principal frequency**, which increases with gray level g. Since noise energy at a significantly higher spatial frequency than f_g is not perceivable, we can optimize a halftoning method for a particular gray level g by forcing the noise spectrum to be concentrated above f_g.

Calculating the noise spectrum. In order to implement the optimization, we first need to compute the noise spectrum. The halftone pattern $x_h(n_1, n_2)$ for the gray level $x(n_1, n_2) = g$ has the error $e(n_1, n_2) = g - x_h(n_1, n_2)$, which is an $N \times N$ image. As explained in [8], a **radially averaged power spectrum** $P_r(k_r)$ for this error is calculated, where k_r is a positive integer called the radial frequency. The class matrix in the dot diffusion method should be optimized such that this radial spectrum is appropriately shaped for a well-chosen fixed gray level g. In terms of the radial frequency variable k_r, the principal frequency for the halftone of gray level g is given by

$$f_g = k_{max}\sqrt{g}$$

where k_{max} is the maximum value of k_r. In fact, for $g > 0.5$, since black pixels are more in number, the halftone is perceived as a distribution of white dots and we have to take $f_g = k_{max}\sqrt{1-g}$.

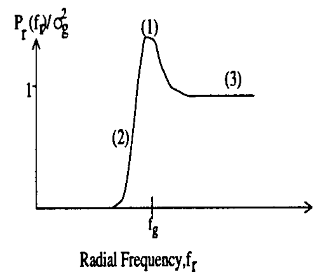

Figure 1: A Typical Desired Radial Spectrum Characteristics

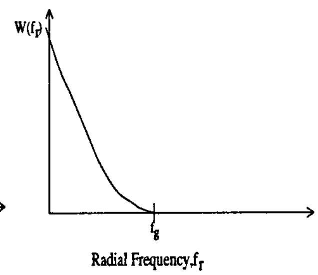

Figure 2: The Weight Function used in the Optimization

The aim of the optimization is to shape $P_r(k_r)$ by choice of the class matrix $\mathbf{C}$ so that most of its energy is moved to the region $k_r > f_g$ (as demonstrated in Fig. 1). We therefore define the cost function

$$\Phi(\mathbf{C}, g) = \int_0^{f_g} P_r(k_r)w(k_r)dk_r$$

The weight function was chosen to be $w(k_r) = (k_r - f_g)^2$ for $0 \leq k_r \leq f_g$ and zero outside [8]. In the optimization the integral was replaced with a discrete sum. The choice of the class matrix that minimizes this sum was performed using the **pairwise exchange algorithm** [8], which was originally proposed in digital filter literature for a different application [1]. The gist of this algorithm is as follows:

1) Randomly order the numbers in the class matrix.

2) List all possible exchanges of class numbers.

3) If an exchange does not reduce cost, restore the pair to original positions and proceed to the next pair.

4) If an exchange does reduce cost, keep it and restart the enumeration from the beginning.

5) Stop searching if no further exchanges reduce cost.

6) Repeat the above steps a fixed number of times and keep the best of the class matrix.

Choice of gray level. Since the algorithm can be applied only to a given gray level, the gray level should be chosen wisely to get good halftones for other gray levels also. For most natural images, the best gray level was experimentally found to be $g = 0.0625$ as explained in [8].

4 INVERSE HALFTONING

Inverse halftoning is the reconstruction of a continuous tone image from its halftoned version. The basic aim in inverse halftoning is to separate the halftoning noise from the original image. In good halftoning algorithms, the noise introduced by halftoning is concentrated in the high frequencies. Simple low pass filtering can remove the high frequency noise but it also removes the edge information. Thus the edge information should be separated from the halftoning noise.

Inverse halftoning using wavelets was considered in [12] and [6]. The algorithm in [12] is tailored for error diffusion, which has different characteristics than dot diffusion. If the method in [12] is used, the result is not good. This can be seen from Fig. 8 which is the result of inverse halftoning of dot diffused Lena by using the method in [12]. The image suffers from periodic patterns, which is essentially low frequency noise.

In the new method, the specific properties of the dot diffusion algorithm are taken into account. The image is enhanced before dot diffusion, hence in the inverse halftoning, the dot diffused image should be deenhanced using the inverse filter of $F_{enh}(z_1, z_2) = 10-(z_1+1+z_1^{-1})(z_2+1+z_2^{-1})$. Note that $F_{enh}(e^{jw_1}, e^{jw_2}) > 0$ for all $0 \leq w_1, w_2 \leq \pi$.

We use the wavelet tree built from the analysis block shown in Fig. 3. An image $C(x,y)$ is decomposed into $L(x,y)$, $H(x,y)$, and $V(x,y)$ using the undecimated wavelet transform. At scale 2^{i+1}, (which will be described below), the filtering operations are as follows :

$$L(w_1, w_2) = F(2^i w_1)F(2^i w_2)C(w_1, w_2),$$
$$H(w_1, w_2) = G(2^i w_1)F(2^i w_2)C(w_1, w_2),$$
$$V(w_1, w_2) = F(2^i w_1)G(2^i w_2)C(w_1, w_2),$$

where G and F are derived from quadratic spline wavelets and they are tabulated with the synthesis filters in Table 1, in [7] (F is H in the latter Table). The choice of filters given in [7] detect edges at different scales if they are used in the wavelet tree shown in Fig. 4 with scales $2^0, 2^1, 2^2, 2^3$ from left to right. For example $H_i(x,y)$ and $V_i(x,y)$ represent the horizontal edges, and vertical edges of $L_{i-1}(x,y)$ at scale 2^{i-1} respectively, and $L_i(x,y)$ is the low pass version of $L_{i-1}(x,y)$.

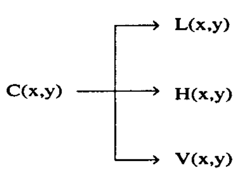

Figure 3: Wavelet decomposition of an image

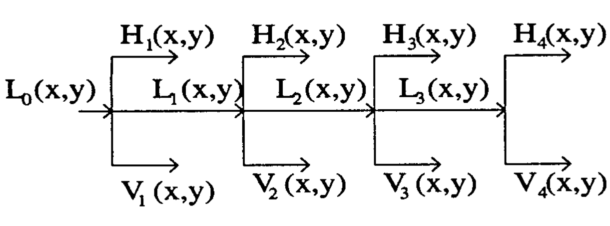

Figure 4: Wavelet Tree used in Inverse Halftoning

Let us denote the ith level low pass image, vertical edge image and horizontal map image as $L_i(x,y)$, $V_i(x,y)$, and $H_i(x,y)$ respectively. The 4-level wavelet decomposition is then applied to the deenhanced halftone image, $L_0(x,y)$. Then for each pixel location (x,y), the following is done:

1) Apply a symmetric FIR Gaussian filter, $f_g(n,m)$ to $V_1(x,y)$, and $H_1(x,y)$. ($f_g(n,m) = ce^{-\frac{n^2+m^2}{2\sigma^2}}$ for $-3 \leq n, m \leq 3$, and c is chosen such that the DC gain of the filter is unity). The first level edge maps contain mostly the halftoning noise, thus low pass filtering these images reduces the blue noise without harming the edges too much.

2) Let $E_{23}(x,y) = V_2(x,y)V_3(x,y) + H_2(x,y)H_3(x,y)$.
if $E_{23}(x,y) \leq T_1$ then make $V_2(x,y) = 0$ and $H_2(x,y) = 0$.

3) Let $E_{34}(x,y) = V_3(x,y)V_4(x,y) + H_3(x,y)H_4(x,y)$.
if $E_{34}(x,y) \leq T_2$ then make $V_3(x,y) = 0$ and $H_3(x,y) = 0$.

Steps 2 and 3 are the denoising steps in the algorithm. In order to discriminate the edges from the halftoning noise, we have to locate the edges. For this, the above steps perform a cross correlation between the edges at different scales. If there is an horizontal edge at scale i at (x,y) then $H_i(x,y)$ and $H_{i+1}(x,y)$ will be of the same sign [7]. The same is also true for vertical edges. Combining the horizontal and vertical edge correlations gives better results in detecting the diagonal edges.

4) The above steps have modified the subband signals L_i, H_i and V_i in certain ways. We now use the inverse filter bank (synthesis bank) corresponding to Fig. 4, and obtain a reconstructed version $\hat{L_0}(x,y)$. The image $\hat{L_0}(x,y)$ is the desired inverse halftone image.

5 EXPERIMENTAL RESULTS

The 512×512 continuous tone peppers image is halftoned by using Knuth's class matrix (Fig 6), and by the optimized class matrix (Fig 7). It is clear that the new method is superior to unoptimized dot diffusion method. In fact, the new method offers a quality comparable to FS error diffusion method (Fig. 5). Error diffused images suffer from worm-like patterns which are not in the original image, whereas dot diffused halftones do not contain these artifacts. Notice that the artificial periodic patterns in Fig. 6 are absent in Fig. 5 and in the new method (Fig. 7).

In inverse halftoning, dot diffusion has an advantage, even the simple unenhanced image is a quite reasonable inverse halftone (psnr=26.62dB for Lena image). The unenhanced image is further processed as described in Sec. 4. The parameters used in the method are found experimentally. The variance of the Gaussian filter, σ^2 is chosen to be 0.5 and the thresholds are chosen to be $T_1 = 300$ and $T_2 = 20$. The results are shown in Fig. 10 (psnr=30.58dB) and in Fig. 9 (psnr=28.61dB). [2]

6 CONCLUSION

Even though dot diffusion offers more parallelism than error diffusion, it has not received much attention. This is partly because the noise characteristics of error diffusion method are generally regarded as superior. We observed that by optimizing the class matrix for blue noise at a fixed gray level, the results of dot diffusion can be made at least as pleasing

[2] The inverse halftone images can be found at htttp://www.systems.caltech.edu/mese/halftone/

as that of error diffusion. The algorithm terminates in at most 64 steps for 8×8 class matrix compared to N^2 steps needed for error diffusion algorithm. Moreover, as noticed in [8], the algorithm can in fact be terminated in about 50 steps. The conclusion is that Knuth's dot diffusion method with a carefully optimized class matrix is very promising; the image quality is comparable to error diffusion, and the implementation offers more parallelism than error diffusion. We also developed a wavelet-based inverse halftoning algorithm which works very well, even though the class matrix information is not used.

References

[1] J.P. Allebach and R.N. Stradling, *"Computer-aided Design of Dither Signals for Binary Display of Images,"* Applied Optics, Vol. 18, pp. 2708-2713, August 1979.

[2] Dimitris Anastassiou, "Neural net based digital halftoning of images," ISCAS, Vol. 1, pp. 507-510, 1988.

[3] B. E. Bayer, *"An Optimum Method for Two Level Rendition of Continuous Tone Pictures,"* in Conf. Rec. IEEE ICC, 1973, pp 26-11-26-15.'

[4] R. W. Floyd and L. Steinberg, *"An Adaptive Algorithm for Spatial Greyscale,"* Proc. SID, pp. 75-77, 1976

[5] D. E. Knuth, *"Digital Halftones by Dot Diffusion,"*, ACM Transactions on Graphics, vol. 6, no. 4, pp 245-273, October 1987.

[6] J. Luo, R. de Queiroz, and Z. Fan, *"A Robust Technique For Image Descreening Based on the Wavelet Transform,"* IEEE Trans. on Signal Processing, vol. 46, no. 4, pp. 1179-1184, April 1998.

[7] S. Mallat and S. Zhong, *"Characterization of Signals from Multiscale Edges,"* IEEE Transactions on Pattern and Machine Intelligence, Vol. 14, No. 7, July 1992.

[8] M. Meşe and P.P. Vaidyanathan, *"Image Halftoning Using Optimized Dot Diffusion,"* Proceedings of EUSIPCO, Rhodes, Greece, 1998.

[9] T. Mitsa and K. J. Parker, *"Digital Halftoning Technique Using a Blue Noise Mask,"* J. Opt. Soc. Am. A, Vol. 9, No. 11, pp. 1920-1929, November 1992.

[10] M. A. Seldowitz, J. P. Allebach, and D. E. Sweeney, *" Synthesis of digital holograms by direct binary search,"* Appl. Opt Vol. 26, pp. 2788-2798, 1987.

[11] R. A. Ulichney, *"Dithering with Blue Noise,"* Proceedings of IEEE, vol. 76, no. 1, pp.56-79, January 1988.

[12] Z. Xiong, K. Ramchandran and M. Orchard, *"Inverse Halftoning Using Wavelets,"* Proc. of Intl. Conf. on Image Processing, Lausanne, CH, vol. I, pp 569-572, 1996.

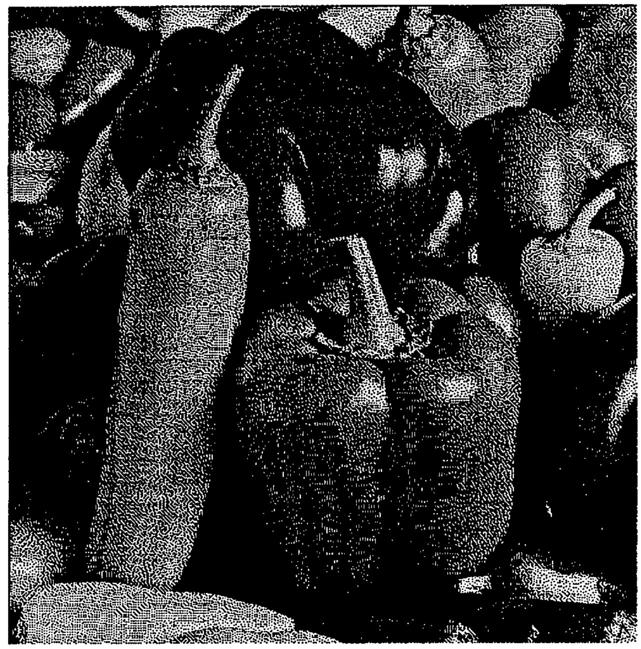

Figure 5: FS Error Diffusion

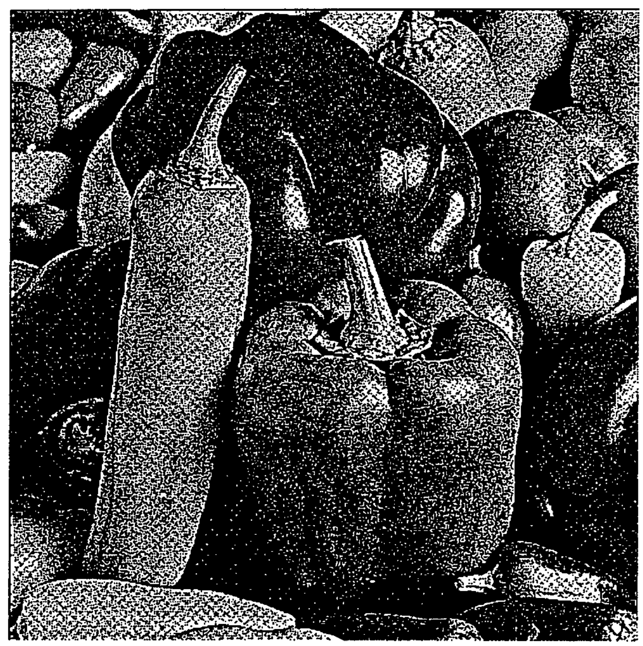

Figure 6: Dot Diffusion with Knuth's Class Matrix

Table 1: Class Matrix **C** for the new method

59	12	46	60	28	14	32	3
21	25	44	11	58	45	43	30
24	20	13	42	33	5	54	8
64	52	55	40	63	47	7	18
35	57	9	15	50	48	4	36
41	17	6	61	22	49	62	34
2	53	19	56	39	23	26	51
16	37	1	31	29	27	38	10

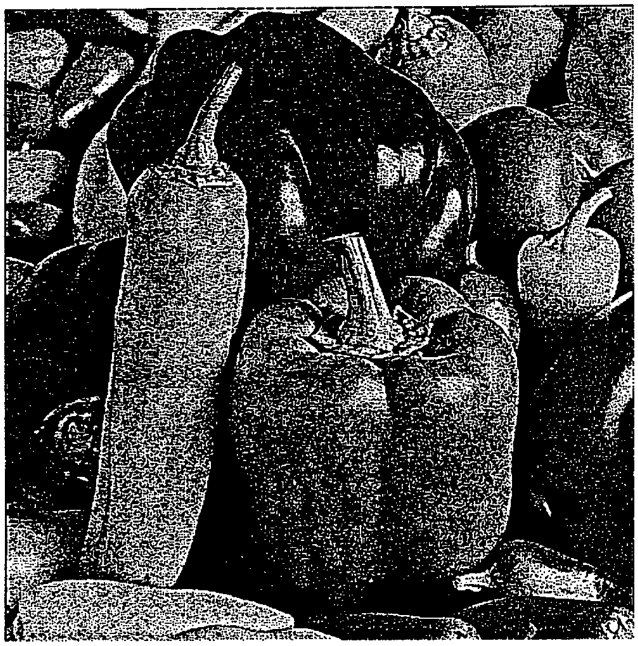

Figure 7: Dot Diffusion with the new Class Matrix

Figure 8: Result of Inverse Halftoning using previous method

Figure 9: Inverse Halftoned Lena Using the new method

Figure 10: Inverse Halftoned Peppers Using the new method